Building Application Services on the Grid: The National Fusion Collaboratory

L. Randerson,[1] J. Burruss,[2] E. Feibush,[1] S. Flanagan,[2] K. Keahey,[3] C. Ludescher,[1] D.C. McCune,[1] Q. Peng,[2] D. P. Schissel,[2] M. R. Thompson[4]

[1]*Princeton Plasma Physics Laboratory, Princeton, NJ*

[2]*General Atomics, San Diego, CA*

[3]*Argonne National Laboratory, Argonne, IL*

[4]*Lawrence Berkeley National Laboratory, Berkeley, CA*

Abstract

The National Fusion Collaboratory is building the Fusion Grid to enable national and global collaborative magnetic fusion research. The goal of the Fusion Grid is access by experimental facilities and multi-institution collaborations to simulation and modeling codes, data, and visualization, using Grid-based services. Experience has been gained through successful deployment of multiple services. This paper describes the infrastructure required to convert computational codes, scientific data collections, and visualization tools into Grid services, and the lessons learned in doing so.

1. Introduction

The National Fusion Collaboratory (NFC) project [1] consists of scientists and researchers from three large fusion experiments and four computer science centers. The experiments are C-Mod [2] at MIT, DIII-D [3] at General Atomics, and NSTX [4] at the Princeton Plasma Physics Laboratory (PPPL). The computer science centers are Argonne National Laboratory, Lawrence Berkeley National Laboratory, Princeton University, and University of Utah. The NFC project is building an interconnected suite of computational applications, visualization, data, and monitoring services for the US plasma physics and fusion research community. This suite of services is known collectively as the Fusion Grid.

Deleted: .

In this fusion research community, typical applications include experiment data analysis, modeling, and simulation of fusion plasmas. The applications are shared and run at many geographically dispersed sites. The Fusion Grid is being built to provide the infrastructure and services needed to convert the present dispersed collection of magnetic fusion computational codes, scientific data servers, and visualization tools into a coherent set of networked services.

Deleted: o

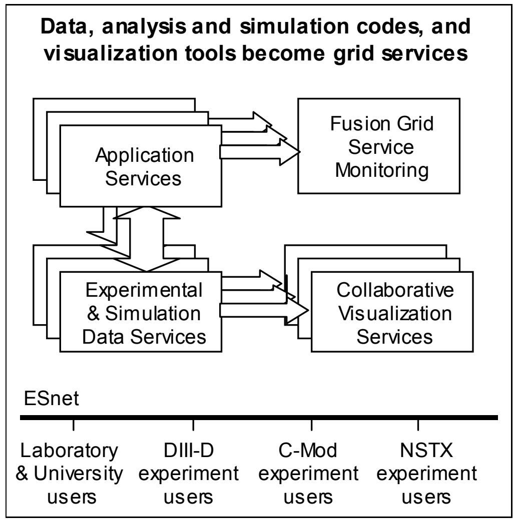

Figure 1. National Fusion Collaboratory services

A complete support structure for the Fusion Grid must achieve several objectives:

- The infrastructure required to use scientific codes, data, and visualization as services.
- Application execution on remote high-performance computers with no loss of functionality or security.
- Increased capability of network interaction of diverse codes and data to enable new capabilities.
- A strong increase in multi-location collaboration.
- Simplification and lower cost of application support for applications used at multiple locations.

To this end, the NFC is building, shown in Figure 1, centralized application services, Grid-wide data services, visualization tools, and monitoring capability. Grid middleware is used for user authentication and job management with per user and per group authorization of service functions.

Grid-wide data services and an important fusion analysis/simulation application service, TRANSP [5], have been placed in production. Useful tools for visualization and monitoring of services exist and are being extended.

The structure of this paper is as follows. Section 2 summarizes the fusion applications computational environment, available Grid middleware, and gives an example of the Grid conversion process. Section 3 describes in more detail the services and tools provided to support the addition of Grid services to the Fusion Grid. Section 4 describes requirements for user authentication and authorization and the procedure for converting an application to a Grid service. Section 5 describes the achievements and lessons learned from building the existing Fusion Grid capability. Section 6 summarizes the results and conclusions.

2. Fusion Applications Environment

Magnetic fusion experiments operate in a pulsed mode, producing seconds-long plasmas every ten to twenty minutes. During and after the pulse, data is acquired and stored in local data servers. Compute intensive codes are used to simulate or model the plasma. The "experimental" codes run for a short time because they must complete before the next pulse to have an impact on experimental operations. The more detailed "theoretical" codes may run for days. Input and results from theoretical codes can be stored in the same data server as the experimental codes but are frequently in an application-specific file format or in a general scientific file format such as NetCDF or Hierarchical Data Format.

The magnetic fusion community shares the computational codes. Typically, one person or site is responsible for improving, debugging, and porting the code to a standard set of platforms; other people and sites then install and use this software on their machines. There are numerous codes in the community. Data requirements for many of these codes are modest: 150 megabytes per run or less. A variety of visualization programs are used for analyzing the results.

The computational codes are a part of rapidly evolving scientific research effort with new features constantly being added. The active engagement of the developer for debugging support is crucial for many of the applications and is highly valued by the users.

Networking speed to user computers within sites is usually 10 Mbps, with hubs joined by 100 Mbps or 1 Gbps networks. Wide area connectivity of sites varies, but is sufficient for networked use of applications with modest data requirements. The high-performance machines available for running fusion codes are typically Linux based Beowulf clusters, although very high performance parallel supercomputers are also available to the community. The cluster job management systems are OpenPBS and PBSPro but could be others.

2.1 Exploiting Existing Grid Middleware

Currently, the Globus Toolkit® [6] and Akenti [7] are the two toolkits used by the Fusion Grid.

- The **Globus Toolkit** is used for authentication, network connections, job submissions, and file transfer. It provides the single sign-on capability through delegation of user credentials. File transfer for Fusion Grid applications uses the Globus Toolkit GridFTP.
- **Akenti**, an access control policy management tool, is integrated with the Globus Toolkit to ensure that a user requesting a function of a service is authorized to request that function or is a member of an authorized group.

2.2 Converting to Applications Services

A server location needs to be selected for applications services targeted for the Grid. In many cases, moving the fusion codes to the Beowulf clusters will significantly speed the computation for both experimental and theoretical codes. Thus the fusion code TRANSP has been deployed as a "normal priority" production service with running times of one or more hours. TRANSP is running on a modern Linux Beowulf cluster at PPPL outfitted with the Globus and Akenti toolkits. This deployment replaced a collection of site-local facilities generally running on older, slower hardware, and has already proven highly successful as an "offline" service for users.

Deleted: The applications considered for becoming Grid data services are compute intensive and simulate or model the plasma.

A future step is to run fast experimental codes (including an optimized TRANSP) as "between pulse" services. These time-critical jobs will require high priority processing with time reservation and preemption capability. Demonstration of such capability is a priority for the computer science community and fusion community research users alike. The NFC has begun carrying out tests to determine how to achieve such performance [8].

Deleted: :

Deleted: 5,

3. National Fusion Collaboratory Services and Tools

Converting magnetic fusion codes to Grid application services requires supporting Grid services in place with the application services that will allow users the same functionality that was provided before Grid conversion. This section discusses the services required, the building

Deleted: other

of application services, client use of these services, and the first steps towards collaborative visualization.

3.1. Grid Monitor and Accounting Service

When an application becomes a service, the tracking of run status and the ability to read a run's log files must be maintained. A new requirement resulting from running on a remote server is that monitoring information must be stored permanently because usage information as to who ran what and how much on the computation server becomes important.

Since the Globus Toolkit's job submission service, GRAM [9], provides only limited monitoring information and the Monitoring and Discovery Service [10] (MDS) lacks subscription capabilities essential for our purpose, the Collaboratory instead chose to implement the **Fusion Grid Monitor** (FGM) [11]. To maintain the permanent database, the FGM uses Java JDBC with a Sybase Linux client to access a Microsoft SQL server. Applications write status information locally and transmit the data regularly to the FGM. If the FGM is not accessible, the information is spooled and transmitted later. The name of the applications log file is sent with the status information and the user can request information from the log file through FGM browsing which fetches the file from the local site by anonymous FTP to display the file contents.

The Fusion Grid Monitor is centered at General Atomics and has been used for a year. The first Fusion Grid application service to use the FGM is the TRANSP application service. In addition to monitoring capabilities, the FGM provides a convenient and portable web browsing, allowing users to view status such as queue start time, computation start time, wall and CPU time, and summary status information such as the time step reached in a simulation. The FGM will be used by each service with the fusion code GS2 [12] expected next.

3.2. Grid Data Service

A Fusion Grid data service should support both experimental and simulation data. The data service must support networked access, be able to handle the scientific data, be easy to use, and easy to set up and maintain at several sites. The system chosen by the Fusion Grid that met these requirements was **MDSplus**.

The MDSplus [13,14] data system was already extensively used by magnetic fusion experimental groups and thus was an obvious choice. MDSplus provides for acquisition, storage, access, and organization of data and can store both experimental and simulation data. It serves data on the networks of the experiment sites and has allowed remote access to specific sites over the Internet since the mid-1990s.

The MDSplus servers and the MDSplus library were Grid-enabled with GlobusIO library routines for authentication and data transfer. The Grid-enabled MDSplus servers are run in parallel with the non-Grid-enabled servers on the same data trees to allow both Grid-enabled and (legacy) non-Grid-enabled applications using MDSplus to access the same data. Applications already using MDSplus can easily become Grid-enabled by rebuilding with the Grid-enabled MDSplus libraries. Applications not already using MDSplus can become Grid-enabled by either modifying their code or, more typically, converting legacy file data from and to MDSplus at the beginning and end of each run.

The distribution of data servers allows decisions as to where the data should reside. Experimental data must be stored and retrieved in a timely matter so it is generally stored at the experiment site. Simulation and modeling data can be stored at the MDSplus server site selected by the community. The per-run amounts of data by the first Grid application services range from 5 megabytes to 150 megabytes; data service transmittal rates at these amounts have not yet been seen as an issue. However, many fusion theory applications have much stiffer requirements, so, high performance network-parallel MDSplus data transfers are being tested.

3.3. Grid Application Services

The codes selected for Grid deployment will be run from a large number of sites. Since the application developer will be doing much of the conversion work, a necessary motivation is that the cost of conversion should be lower than the cost of maintaining the code at many sites. Of course, once a set of application services has been deployed, a future advantage of Grid service deployment will be that it allows automated collaborative interconnection of services and efficient sharing of geographically distributed fusion community resources.

Specific tasks faced by the Fusion Grid application service developer Fusion Grid include deciding which services should be available, writing scripts that can execute these services on the local compute service hosts, converting the application to read/write to the MDSplus data server (or provide pre-processing and post-processing tasks to do this), deploying the Globus Toolkit, and providing commands to the clients to invoke these services over the Grid.

Clients must install the Globus Toolkit software and obtain X509 certificates from the DOEGrids Certificate Authority [15]. After logging on to the Fusion Grid by getting a (pass-phrase protected) certificate proxy, clients can then request the application services. MDSplus-enabled tools are subsequently used to access the results of the run.

Deleted: of

Deleted: and

Deleted: The second will be the GS2 [12] application service.

Deleted: (

Deleted:)

3.4. Grid Application Utilities

Once Fusion Grid services have been deployed, Grid application clients can adapt their input preparation, application execution, monitoring, and output analysis to use these services.

The first applications to become Fusion Grid services are TRANSP, which provides time-dependent modeling for magnetic fusion energy experiments, and GS2, a parallelized physics application code developed to study low-frequency turbulence in magnetized plasma.

For the TRANSP application, the GUI utility PreTRANSP [16], shown in Figure 2, was built. PreTRANSP prepares input data and submits requests to start a TRANSP run. The utility loads selected input data files into an MDSplus data service and submits a request to execute the TRANSP application along with an input file defining the run id and the location of the MDSplus data service to use for input and output data. The submit request includes a time-limited DOEGrids certificate proxy for authentication of the user.

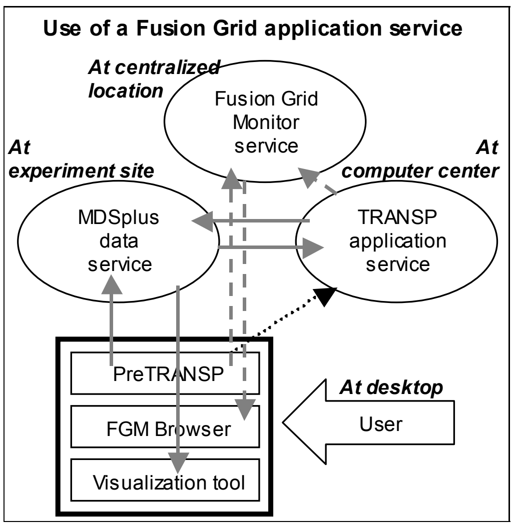

Figure 2. Using PreTRANSP utility to access services

As the request is being submitted, the PreTRANSP utility notifies the Fusion Grid Monitor service. TRANSP itself sends updates to the FGM during execution. Users can follow run status from a browser by viewing the FGM output.

3.5. Collaborative Visualization

The creation of Fusion Grid MDSplus data service-enabled visualization tools was the first step toward collaborative visualization. Geographically distributed co-workers can all access and graph the same data each with their own copies of the tools. Sharing these graphs quickly and conveniently across the network for discussions is the next step.

Currently under development is ElVis [17], based on Scivis [18], which is a client-server visualization system allowing the collaborative sharing of graphs over the Grid. The ElVis system consists of data source clients, servers, and display clients. Data source clients create graph objects sent to the ElVis server. Display clients register with the server the data sources they want to receive. The server forwards the graph objects to the registered display clients. Display clients can form collaborative groups and exchange graphs. All clients can add, delete, and manipulate the shared graphs. Changes are instantly and efficiently updated in all collaborating display clients.

TRANSP RPLOT, a tool for reading and plotting the results of completed TRANSP runs, can now act as an ElVis data source. The TRANSP application itself can be a data source client updating plots once per minute for a lengthy run. These ElVis deployments, which have run as prototypes in the NFC project, allow both for interactive analysis of completed TRANSP runs and for the remote graphical monitoring of the physics results of the runs as they execute.

4. Converting a Computational Application Code to a Grid Service

In this section, the issues involved in converting a magnetic fusion computational code into a Grid service are considered in detail.

4.1. Application Service Authorization

Akenti provides access control integrated with the Globus Toolkit to ensure that a user who has requested a service is authorized. For the Fusion Grid, Akenti can allow levels of access for each type of application user. For example, Akenti groups can be defined as "Developers." "Clients," and "General." Developers can run all services. Clients can run a more restricted predefined set of normal application user services such as start, look, halt, and archive. General users can test access to the system by allowing execution of minimal programs such as `/bin/sleep` and `/bin/date`.

Application users are identified to the Akenti database by the same DOE Grids X509 certificates used for authentication. The Globus Toolkit job manager uses Akenti authorization after authentication has been performed by the gatekeeper. The user will receive a failure notice at submit time if requesting a procedure that he or she is not authorized to use.

Deleted: be

Deleted: With the

Deleted: is

Deleted: results

Deleted: After

Deleted: has been

Deleted: At the end of the run, visualization tools access the data server to display run results.

Deleted: checks

Future Akenti functions will allow group leader control of jobs submitted by members of the leader's group. This approach will allow monitoring of runs and the necessary control of runs by group leaders.

4.2. Building an Application Service

The following steps are needed to create a Grid-enabled application service:

- A Grid-enabled computer must be provided and the application must be ported to it.
- User (client) methods for accessing the application have to be updated.
- The application program has to be changed to either use the Grid enabled data server or to convert files from and to the data server.
- User analysis and visualization programs need to be updated to get application results from the data server.

These requirements are not Fusion Grid specific, but generic to all Grid applications.

To be Grid enabled, the computer site where the application service is running must install Globus Toolkit utilities and libraries, the Globus Toolkit job manager, and the GridFTP server. These can be downloaded for installation from the Globus Project website [19]. From the DOEGrids certificate authority; X509 certificates for the hosts running the job manager and GridFTP must be obtained, along with signing policy files for user certificates [15]. Moreover, DOEGrids for each user must be obtained and the host Globus user database updated.

To access the application service, client sites must install the Globus Toolkit client software plus the Grid-enabled MDSplus libraries. Both of these packages can be downloaded from Linux binaries on the Globus download site. To access Fusion Grid services including application services, each user must obtain a DOEGrids user certificate. Application websites should be updated to guide users through this procedure.

When planning the application conversion to a service, the developer must decide the functions to control the application (such as Start, Stop, and Return-Status. Shell scripts are written to invoke them from the Globus job manager and user scripts to request them. Optionally, an Akenti authorization policy to determine the privileges of each user (i.e., which functions they can invoke) will be specified in an Akenti database. Akenti can be installed as a Globus Toolkit add-on during the Globus Toolkit computer site installation. The developer may also have to determine which data server is to be used, although this can also be left for user specification.

If the application already uses the MDSplus data server locally, then it can be rebuilt with the Grid-enabled MDSplus library. If not, then either the application can be modified to use MDSplus directly, or the scripts invoking the application can be modified to run ancillary programs to convert from MDSplus to the application input file before application start and from the application output file to MDSplus after finish.

Fusion Grid status sending should done at start time and at finish time. A long running job should regularly send information gathered from the log file or other means to mark the progress of the simulation; this information is sent to the Fusion Grid Monitor via a Java interface.

Moreover, to support user analysis of results, analysis and visualization programs must be updated to get results from the Fusion Grid MDSplus data server.

To do these tasks, the application developer does not need systems level Grid software knowledge. The internals of the application do not have to be changed. The possible change of results analysis and visualization programs to get data from the data server may require some effort, but the resulting ability to exchange data with other applications over a network should be a significant benefit.

5. Achievements, Issues, Concerns, and Lessons Learned

The National Fusion Collaboratory is finishing its second year. Sufficient experience with support of computational applications services and data services is in place to allow a retrospective analysis.

5.1. Achievements

The infrastructure required to use scientific codes, data, and visualization tools as Grid services has been achieved in the Fusion Grid. The Globus Toolkit integrated with Akenti access control has been used as Grid middleware. A Grid monitor allows tracking of application runs during execution. The Fusion Grid MDSplus data service provides a common data source used by applications for data input and storage and for access by visualization tools during analysis.

The goal of execution of applications on remote high-performance computers, with no loss of functionality or security, has been obtained. Globus Toolkit user authentication and Akenti access control provide Grid security. Grid services for application monitoring and data serving allow remote applications to maintain the same level of information feedback to users during execution as did earlier system implementations based on purely local resources.

Deleted: group leaders

Deleted: possible

Deleted: the run

Deleted: a

Formatted: Bullets and Numbering

Deleted: <#>User (client) methods for accessing the application have to be updated.¶
¶
<#>The application program has to be changed to either use the Grid enabled data server or to convert files from and to the data server.¶
¶
<#>User analysis and visualization programs need to be updated to get application results from the data server.¶

Deleted: a

Deleted: certificate to the computer site user account-mapping file must be updated with users to access the application service.

Deleted: To access the new application service, the application developer must provide new scripts and MDSplus data server-enabled analysis programs.

Deleted: determined and

The deployment of the first application service, PPPL's TRANSP, has had important measurable benefits. Deployment has been simplified to one centralized site instead of multiple differing sites. Users of this code have seen the expected significantly better throughput. Sites at General Atomics and MIT report a labor savings of several man-months per year since local maintenance of this large, complex, evolving research code is no longer required. Developers can now track runs quickly through the Fusion Grid Monitor and can troubleshoot problems at a centralized site with a known version of the code in a familiar programming environment.

The efficacy of the MDSplus data service has been demonstrated. The conversion of applications on the Grid to use a common data service is expected to increase the use of multiple codes to analyze problems. For example, it is expected that the TRANSP application will use the GS2 theory code Grid service to access a first-principles physics based transport model for predictive modeling.

Grid-enabled collaborative visualization and graphical monitoring of application results, while in an early stage of development, has been demonstrated using the ElVis collaborative visualization software.

Fusion community acceptance of the first Fusion Grid application service (TRANSP) has demonstrated the advantages and cost savings of moving computational applications from multiple productions systems installed and run at many sites to a Grid service at a centralized site available to all. This beneficial experience will be leveraged to deploy additional Fusion Grid services.

5.2. Issues, Concerns, and Lessons Learned

A natural next step is enabling computational application services to access other computational application services. Many fusion simulation programs can take each other's data as input. For example, fusion experimentalists use the results of completed TRANSP runs to drive the GS2 theory model to study turbulent transport in fusion plasmas. This brings up the issue for the correct paradigm for application services used this way. Should MDSplus servers be used for all data communication or do communication, or do less expensive methods exist for sharing "temporary" data or performing secure "task-to-task communication" between applications services?

Another issue is multi-site scheduling. Support will be necessary for the coordinated use of multiple application services across the Grid. This scheduling cannot be done at the local level because the coordinated services may reside in separate administrative domains.

Data access support may prove to be a major hurdle. In an environment such as the magnetic fusion community, multiple non-Grid codes at each site reference each other's data through a variety of methods. What should be the data access support for the non-Grid enabled applications once the Grid applications begin using a common data server? Our current approach is to use a data server that can be accessed either by the Grid or locally. This resolves the issue for applications already using the data server. Other applications resolve the issue by writing both to the data server and to their old file format but such an approach may raise the issue of supporting a file distribution service.

If a central monitoring service (the Fusion Grid Monitor) is to retain persistent information on services for accounting purposes, this requires assured transmission of the status information from the application to the monitor, with spooling for later transmission if the monitor or network connection is down. The NFC project's first attempt at building the monitor used the Grid middleware's information sending service, which sends information such as CPU type and load status regularly to a server on the monitor side. This information was settable and allowed the definition of application information that an application could send. However, since there was no verification when messages were received, it was difficult from the application side to know when to resend the message. This issue led the NFC to alternative monitoring communications methods.

Deleted: as

Deleted:

Deleted: HE

Another issue is control and support of applications services. Traditionally, applications developers have had enough privileges on local operating systems on which the codes were run to halt jobs, to clean up directories, or to perform other control functions on behalf of users. Thus, the developers were able to resolve error conditions, control runaway jobs, and handle other diverse problems. Once an application and data servers are on the Grid, however, the developer is just another user, with no special privileges to resolve problems. This problem motivated enabling Grid-wide job management authorization for the Fusion Grid, as implemented by the Globus Toolkit and Akenti. Once we have installed this feature, it is expected that the problem of applications control will to be resolved.

Another issue is access to non-owned data from a credential-protected Data server access for debugging purposes by the application developer or code support person dealing with a crashed run on behalf of the user and owner of the data. This has been a usage issue in the Fusion Grid and may have to be resolved by defining a means for person-to-person delegation of restricted capabilities.

The major issue encountered during deployment of Grid services has been the firewall policy of each site. Access to services requires the permission of the sites providing the services and for some locations the permission of the site requesting the service with these rules being enforced through firewall policy. Negotiating with network security authorities has had to be done on a site-by-site basis. This has had the practical effect of greatly increasing the startup difficulty of a client at each new site. Problems have occurred on both client and on the server side. Prompted by the prevalence of these problems, the Globus Toolkit developed a firewall "best practices" guide [20], and the Collaboratory is considering recommending a common firewall policy at fusion community sites. However, expected extension of National Fusion Grid services to the international fusion community will likely exacerbate difficulties with disparate site firewall policies for some time to come.

A final issue that will need to be addressed is network transfer speed, which can limit the amount of data transferred. This factor will become a more important consideration as more applications with large data requirements become services. The MDSplus data service is improving the speed of large data transfers, with the objective of allowing application services with larger data amounts to be used more effectively.

6. Summary

This paper examined the process of converting computational applications to services in the Fusion Grid. The environment examined consists of many geographically distributed sites with independent developers each building codes on their home site and then distributing the codes to other sites for installation and use. These computational codes are often long running and computationally intensive.

Application services on the Grid allow the codes to be run at centralized sites with powerful computers, increasing both throughput capacity and the capability of networked interaction of diverse codes and data to enable new combinations of services, all with centralized deployment simplifying the aggregate support effort.

To enable the building of applications on the Grid, a Grid infrastructure has been created with capabilities such as Grid data services and an application Grid monitor. Security on the Grid is addressed with certificate-based authentication. The prototype collaborative ElVis visualization tool has demonstrated multi-site use and collaborative display of analysis results.

The first application services have been built and are serving as prototypes for other application services. The three major U.S. fusion experiments are using the Fusion Grid TRANSP application service to produce their TRANSP runs. The success of the TRANSP deployment has enabled legacy production systems at all sites to be retired, at considerable cost savings.

The first application services are based on high-performance computer clusters and are demonstrating far more powerful analysis throughput capacity with better expert support than the scattered older local systems that were replaced. Grid conversion has not necessitated rewrites of applications and the Grid enabling has been of modest cost to the developer. With the success of the first application services, more applications have been selected to become Grid services. The success of the Fusion Grid should encourage the use of Grid technology generally.

Deleted: and

Acknowledgment

This project was supported by the U.S. DOE Office of Science under the Scientific Discovery through Advanced Computing initiative (SciDAC).

References

[1] K. Keahey, T. Fredian, Q. Peng, D.P. Schissel, M. Thompson, I. Foster, M. Greenwald, and D. McCune, "Computational Grids in Action: The National Fusion Collaboratory", Future Generation Computer Systems, 18:8, pg. 1005-1015, October 2002.

[2] I. Hutchinson et al., Alcator C-Mod., Phys. Plasmas 1 (1994) 1511

[3] J.L. Luxon et al.,in: Proceedings of the 11th European Conference on Controlled Fusion and Plasma Physics, Aache, Federal Republic of Germany; European Physical Society, 1983

[4] S. Kaye et al., Physics design of the NSTX, Fusion Technol. 36 (1999) 16-35

[5] TRANSP, http://w3.pppl.gov/transp/.

[6] I. Foster and C. Kesselman. Globus: A Metacomputing Infrastructure Toolkit. International Journal of Supercomputer Applications, 11 (2). 115-129. 1998

[7] M. Thompson, A. Essiari, K. Keahy, V, Welch, S. Lang, and B. Liu, Fine-grained authorization for job and resource management using Akenti and Globus, 2003 Conference for Computing in High Energy and Nuclear Physics (La Jolla, CA, March 24-28, 2003) http://www.fusiongrid.org/research/papers.html

[8] K. Keahey and K. Motawi, Taming of the Grid: Virtual Application Services.2003, Argonne National Laboratory, Mathematics and Computer Science Division Technical Memorandum ANL/MCS-TM-262

[9]Czajkowski, K., et al., A Resource Management Architecture for Metacomputing Systems, in 4th Workshop on Job Scheduling Strategies for Parallel Processing. 1998, Springer-Verlag. p. 62-82.

[10] Allcock, W., et al. Secure, Efficient Data Transport and Replica Management for High-Performance Data-Intensive Computing. in Mass Storage Conference. 2001.

[11] Q. Peng, J. Burruss, S. Flanagan, K .Keahey, S. Lang, C. Ludescher, D. C. McCune, L. Randerson, D. P. Schissel, and M. R. Thompson, National Fusion Collaboratory: Building the Fusion Grid, Presentation at GlobusWorld, San Diego, California, January 2003, http://www.fusiongrid.org/research/papers/globusworld2003.pdf

[12] GS2, http://gs2.sourceforge.net.

[13] J.A. Stillerman, T.W. Fredian, K.A. Klare, and G. Manduchi, MDSplus data acquisition system, Rev. Sci. Instrum. 68 (1997) 939-942.

[14] MDSplus: Introduction, http://www.mdsplus.org.

[15] DOEGrids CA, http://doegrids.org

[16] Using PreTRANSP at DIII-D, http://fusion.gat.com/comp/analysis/grid/pretransp.html

[17] ElVis, http://w3.pppl.gov/~efeibush/elvis

[18] B. Ki and S. Klasky, Scivis, Concurrency Practice and Experience 10, 1107-1115

[19] The Globus Project, http://www.globus.org

[20] V. Welch, Globus Toolkit firewall requirements, http://www.globus.org/security/v2.0/firewalls.html

Government License

The submitted manuscript has been created by the University of Chicago as Operator of Argonne National Laboratory ("Argonne") under Contract No. W-31-109-ENG-38 with the U.S. Department of Energy. The U.S. Government retains for itself, and others acting on its behalf, a paid-up, nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government